0. Hack the VM

Write a script that finds a string in the heap of a running process, and replaces it.

- Usage: `read_write_heap.py pid search_string replace_string`
  - where `pid` is the pid of the running process
  - and strings are ASCII
- The script should look only in the heap of the process
- Output: you can print whatever you think is interesting
- On usage error, print an error message on `stdout` and exit with status code `1`

Terminal 1:

```
┌──(maroua㉿HBTN-LAB)-[~/0x04. Buffer overflow]
└─⚑ cat main.c
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <unistd.h>

/**
 * main - uses strdup to create a new string, loops forever-ever
 *
 * Return: EXIT_FAILURE if malloc failed. Other never returns
 */
int main(void)
{
    char *s;
    unsigned long int i;

    s = strdup("Holberton");
    if (s == NULL)
    {
        fprintf(stderr, "Can't allocate mem with malloc\n");
        return (EXIT_FAILURE);
    }
    i = 0;
    while (s)
    {
```

```
        printf("[%lu] %s (%p)\n", i, s, (void *)s);
        sleep(1);
        i++;
    }
    return (EXIT_SUCCESS);
}
┌──(maroua㉿HBTN-LAB)-[~/0x04. Buffer overflow]
└─⚑ gcc -Wall -pedantic -Werror -Wextra main.c -o main
┌──(maroua㉿HBTN-LAB)-[~/0x04. Buffer overflow]
└─⚑ ./main
[0] Holberton (0x555e646e02a0)
[1] Holberton (0x555e646e02a0)
[2] Holberton (0x555e646e02a0)
[3] Holberton (0x555e646e02a0)
[4] Holberton (0x555e646e02a0)
[5] Holberton (0x555e646e02a0)
[6] Holberton (0x555e646e02a0)
[7] Holberton (0x555e646e02a0)
[8] Holberton (0x555e646e02a0)
[9] Holberton (0x555e646e02a0)
[10] Holberton (0x555e646e02a0)
[11] Holberton (0x555e646e02a0)
[12] Holberton (0x555e646e02a0)
[13] Holberton (0x555e646e02a0)
[14] Holberton (0x555e646e02a0)
[15] Holberton (0x555e646e02a0)
[16] Holberton (0x555e646e02a0)
[17] Holberton (0x555e646e02a0)
[18] Holberton (0x555e646e02a0)
[19] Holberton (0x555e646e02a0)
[20] Holberton (0x555e646e02a0)
...
[78] Holberton (0x555e646e02a0)
[79] Holberton (0x555e646e02a0)
[80] Holberton (0x555e646e02a0)
[81] Holberton (0x555e646e02a0)
[82] Holberton (0x555e646e02a0)
[83] Holberton (0x555e646e02a0)
[84] Holberton (0x555e646e02a0)
[85] Holberton (0x555e646e02a0)
[86] Holberton (0x555e646e02a0)
[87] Holberton (0x555e646e02a0)
[88] Holberton (0x555e646e02a0)
```

```
[89] Holberton (0x555e646e02a0)
[90] Holberton (0x555e646e02a0)
[91] Holberton (0x555e646e02a0)
[92] maroua (0x555e646e02a0)
[93] maroua (0x555e646e02a0)
[94] maroua (0x555e646e02a0)
[95] maroua (0x555e646e02a0)
[96] maroua (0x555e646e02a0)
[97] maroua (0x555e646e02a0)
[98] maroua (0x555e646e02a0)
[99] maroua (0x555e646e02a0)
[100] maroua (0x555e646e02a0)
[101] maroua (0x555e646e02a0)

...
```

Terminal 2:

```
┌──(maroua㉿HBTN-LAB)-[~/0x04. Buffer overflow]
└─⚑ ps aux | grep ./main
maroua     6515 0.0  0.0   2776  1040 pts/1    S+   10:53   0:00 ./main
maroua     6575   0.0  0.0   9220  2344 pts/2    S+   10:53   0:00 grep --
color=auto ./main
┌──(maroua㉿HBTN-LAB)-[~/0x04. Buffer overflow]
└─⚑ sudo python3 ./read_write_heap.py 6515 Holberton "maroua"
```